REAL TIME DETECTION OF TRAFFIC SIGNS

A Thesis
Presented to
The Academic Faculty

by

EL HAFIDI, BADR

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2018

REAL TIME DETECTION OF TRAFFIC SIGNS

Approved by:

Dr. Tsai, Yi-Chang, Advisor
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. Yezzi, Anthony
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Wang, Zhaohua
School of Civil and Environmental Engineering
Georgia Institute of Technology

Date Approved: December 6, 2018

Dedicated to my mother, my father, my wife and my family,

who offered unconditional love and support and have always been there for me.

Thank you so much.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Dr Tsai, Yi-Chang for his support, his patient guidance and his wise advice throughout my master thesis. The door to Dr Tsai's office was always open whenever I had a question or ran into an issue concerning my research. I would also like to thank my committee members Dr. Yezzi, Anthony and Dr. Wang, Zhaohua. Your willingness to give your precious time has been greatly appreciated. In addition, a special mention goes to Wu, Yi-Ching for her valuable support and engagement, and for her help in providing me with the data I needed for my research work. Similarly, I would like to acknowledge every Professor or colleague I met during my journey at Georgia Tech, who has closely or remotely contributed to the work presented in this thesis. Last but not least, I must express my profound gratitude to my parents, my wife and my family for their endless support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

C

CMYK
Cyan, magenta, yellow and black ..... viii, 2, 3

G

GPS
Global Positioning System 1
GPU
Graphics processing unit 40

H

HSV
Hue Saturation Value ..... xi, 4, 5, 8, 10, 11, 12, 13, 14, 15, 16, 18

M

MBR
Minimum bounding rectangle 24, 25, 26

R

RGB
Red Green Blue 1, 2, 4, 5, 6, 8, 9

SUMMARY

The objective of this research project is to develop a high-performance real-time algorithm for traffic sign detection using a colorful image. The detection of traffic signs is achieved through two steps: the first step extracts the regions of the same color as the traffic signs using an improved color segmentation filter and the second step identifies the shape of the extracted region using a shape recognition algorithm. Classic color segmentation approaches use only one color-range. While they work properly with good weather conditions and high-quality images, they usually struggle with illumination changes, shadow, highlight and harsh weather conditions. Moreover, they can't extract the traffic sign from the background if the latest is in the same color range of the traffic sign. The proposed work outlines an approach that combines both shape and color and uses an improved color segmentation algorithm. It uses a specific color space, which is HSV space, and then builds a set of HSV domains that each corresponds to a specific lightening condition. The approach developed in this work demonstrated the invariance of the detection to shadow, light conditions and weather conditions.

CHAPTER 1 INTRODUCTION

1.1 The large project

The thesis research is part of a larger project that aims to detect, recognize and estimate the 3D position of traffic signs. A smart phone, attached to the windshield of a Van, is used to record a video and GPS locations. The video and GPS data are then processed to extract frames and their correspondent GPS coordinates. The detection algorithm detects the traffic signs of each frames and save the bounding boxes. A structure from motion algorithm is then used to estimate the position of each traffic sign and a neural network is used for classification. The performance of the detection phase is crucial since the performance of the whole processing pipeline depends on it. The detection algorithm is designed to generate the maximum number of true positives; thus, it should minimize the false negatives as much as possible. The number of false positives should not be extremely high, but it is less of an issue at this stage of the pipeline because they will be filtered in the recognition phase after the detection.

1.2 Review of color models

A color model describes colors in a 3D or 4D space as a mathematical model. Thus, a color can have three or four components. Each color model defines how each component should be interpreted considering human's perception of colors. There a different color models and each one is used for specific image processing applications.

1.2.1 RGB color space

The RGB color model represents each color as tuples of additive primary colors Red, Green and Blue. The RGB model is very close to the human vision interaction with

light. The human eye has three cone cell types S, M and L. The Red, Green and Blue light stimulates respectively the L, M and S cone type. Since the three types of cones are sensitive to blue, green and red, this color model is used in light transmitting media such as screens. To achieve a good human color experience only a mixture of red, green and blue light should be generated to cover a large portion of the human color space.

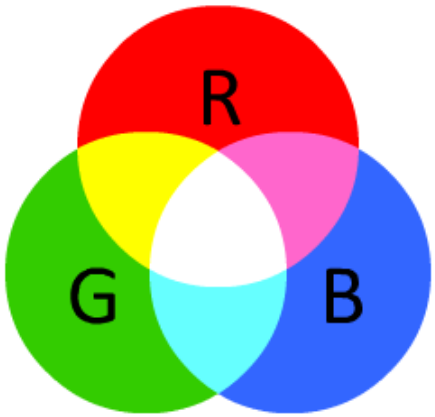

Figure 1: The additive color mixing of the RGB color model

1.2.2 CMYK color model

The CMYK color model describes each color as a combination of cyan (C), magenta (M), yellow (Y) and black (K) color. It is the color model used by printers. The use of the subtractive primary colors cyan, magenta, yellow and black as inks printed on a white paper can reproduce a large portion of the human color space.

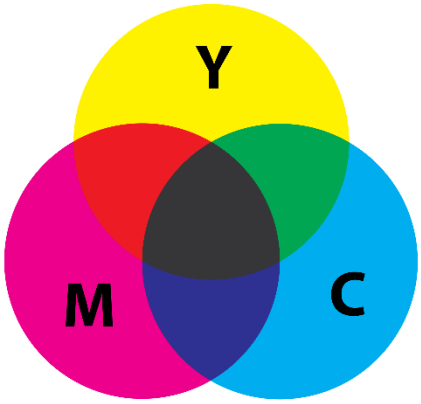

Figure 2: The subtractive color mixing of the CMYK color model

1.2.3 L*a*b* color model

The L*a*b* color model, also called the CIELAB color model, is a three-dimensional model specified by the International Commission on Illumination and describes colors as tuples of lightness (L), position between red/magenta and green (a*) and position between yellow and blue (b*).

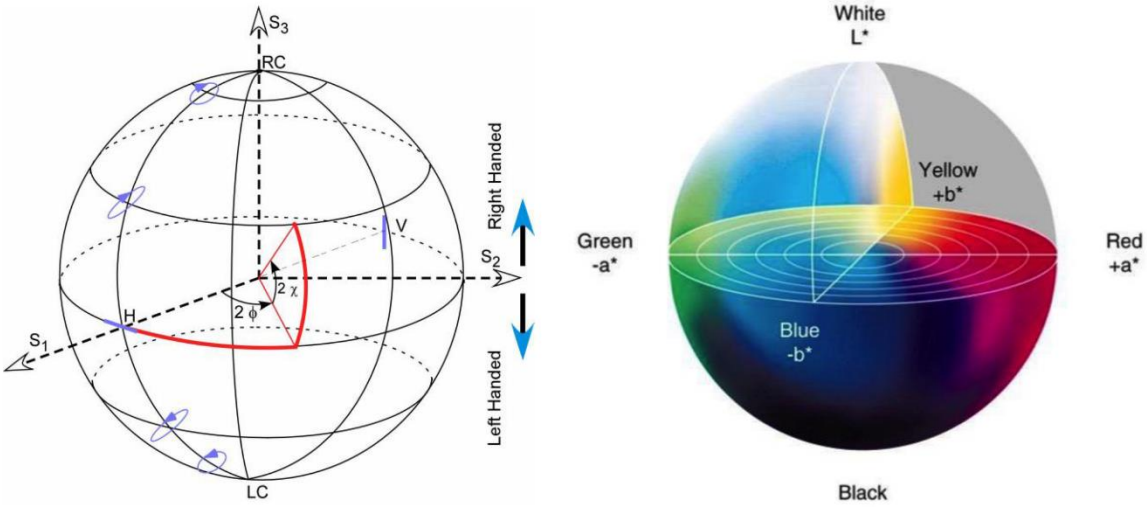

Figure 3: L*a*b* representation in spherical coordinate system ((left), from [1]) and L*a*b* color sphere (right) (from [2])

1.2.4 HSV color model

The HSL, HSB or HSV (Hue Saturation Lightness / Brightness / Value) color model was developed in mid-1970 by computer graphics pioneers in order to create an intuitive color model. HSV color model is very intuitive. The hue is the color portion of the color model and it ranges from 0 to 360 degrees. The saturation describes the amount of grey in the color and it ranges from 0 to 1 or 100%. A low saturation value produces a faded effect and introduces more grey to the color. The value describes the brightness and intensity of the color, and it ranges from 0 to 1 or 100%.

1.3 Review of literature

There are three approaches to detect traffic signs. The first one is based on colors from RGB image and uses several techniques to find the traffic sign in the image, the second approach uses the traffic shapes only and searches for a specific shape in a grayscale image, the third approach uses both color and shape detection.

1.3.1 Detection of traffic signs using shape

The shape-based approach can be very useful since it only uses grey scale images, so it avoids all the problems related to colors like illumination change due to occlusion or low light at night. However, it needs to be translation, rotation and scale invariant. In the following papers, robust techniques have been developed to extract traffic signs using only shapes.

Huang and Hsu [3] used a Matching Pursuit method to design a fast algorithm that extracts circular or triangular shapes form a region of interest. The template matching is

used in this step. The detection rate was at 95% for circular signs and 93% for triangular signs.

Sandoval, Hattori, Kitagawa and Chigusa [4] developed a road sign detection system based on edge detection. This method generates a convolution mask that depends on the angle and position of each pixel. The mask filters the edges that do not fulfill the desired conditions. This method is implemented to detect circular shapes.

1.3.2 Detection of traffic signs using color

The color-based approach takes advantage of the diverse information in a colorful image to extract traffic signs. The color-based algorithms usually require less computational resources, but they struggle with occlusions and illumination changes. The following papers outline efficient color-based methods to detect traffic lights.

Ghica, Lu and Yuan [5] used a reference color to define background pixels and object pixels using threshold segmentation in RGB space. This is achieved by computing the distance between the pixel color and the reference color, if the distance is smaller than the threshold then this pixel is labeled as an object pixel.

Shadeed, Abu-Al-Nadi and Mismar [6] used the YUV and HSV color space to detect road signs. The detection is done within two steps, in the first step the image is projected into YUV color space and a new RGB image is built after equalizing a histogram of the Y dimension, in the second step the new RGB image is now projected to HSV and YUV color space. The detection is achieved by applying a threshold to H and UV values and then combining the two results.

Fang, Yen, Fuh and Chen [7] used the HIS color space in a computational model (dynamic visual model) inspired by human cognitive processes for detecting changes of driving environment. The detection is achieved by computing the hue value of each pixel then the similarity between this hue value and hue values of specific colors. Finally, a perceptual analyzer uses the best degree of similarity to specify the color of the sign.

1.3.3 Detection of traffic signs using both shape and color

Using only shape information or only color information for detecting traffic sign means that we are losing valuable information. Shape based detection methods and color-based detection methods face both different challenges. Therefore, developing a detection system that is based on both shape and color will overcome the constraints related to one approach or the other. The following papers are an example of how shape and color can be used in a hybrid detection system for traffic signs.

Ohara, Nishikawa, Miki and Yabuki [8] carried out detection using two neural networks based on both shapes and colors. The algorithm starts by applying a Laplacian of Gaussian filter on the RGB image to get relevant area, then a color neural network classifies each pixel whether it has the same color as road signs or not, finally a shape neural network classifies each region from the relevant area whether it has a shape of a road signs or not. Template matching is used in the second stage for recognition.

Miura, Kanda and Shirai [9] used both shapes and colors to detect traffic signs. The algorithm works on speed signs, unlike many other methods, it doesn't detect the red boundary region but the white circular region since it has higher intensity values. The white

circular regions are detected using binarization with area filtering and the threshold is determined by analyzing the distribution of data in the YUV color space. Then shape information is used for screening candidates detected in the previous step.

1.3.4 Discussion and remarks

From this literature review, it can be noticed that shapes and colors are valuable information for detecting traffic signs. They are very efficient when they are used under the appropriate circumstances; however, they struggle when they are used under poor conditions. Therefore, a system that can use the right approach depending on the current running condition can overcome the difficulties related to the poor conditions of each method. As noted from the literature review, a lot of work has been done for detecting traffic signs and each approach offer several techniques that are very robust and efficient, but no one emerged as a clear winner. In the following section, the commonly used techniques and their difficulties will be summarized.

The shape-based approach usually works on grey scale images and the edges of the traffic signs. There are three techniques commonly used in this approach, Hough transform, similarity detection and distance transform shape matching. The Hough transform is a simple technique based on a voting procedure to extract features of certain shapes like lines and circles. This method needs, however, a lot of memory and computational resources especially for feature space of more than three dimensions. Similarity detection technique tries to match a segmented region to some samples based on the similarity factor. The distance transform shape matching is based on a distance map which is a representation of the image where each pixel's value is the distance to the nearest obstacle pixel. The

advantage of this method is its ability to work on arbitrary shapes without significantly increasing the computational complexity. The shape-based method has troubles implementing scale and rotation invariant detecting system, it also struggles when the sign is small in the image or when shapes similar to traffic signs exist in the image. This method usually uses algorithms that need high computational resources when working on the whole image.

The color-based approach works on RGB images. Some researchers work directly with the RGB image and others prefer to transform it to other space color like HSV, HIS, HLS, HSB, L*a*b, YIQ or YUV. Those color spaces are more immune to brightness changes than the classic RGB space. One of the commonly used methods is color segmentation; it classifies pixels using a threshold and can be used on any color space. Sometimes it uses a dynamic threshold depending on environment's parameters. Ones of the interesting color spaces are those who extract the hue and saturation from the pixels value like HSV/HIS spaces. Color indexing is also a good and fast technique based on color histograms comparison. The color-based approach struggles working under low light conditions, snow fall or heavy rains. It also has troubles working on traffic sign with different brightness due to shadows. Therefore, an improved color segmentation algorithm should be developed to overcome some difficulties related to shadow, highlight and weather conditions.

CHAPTER 2 ADVANCED COLOR SEGMENTATION

2.1 The right color model for traffic sign detection

The design of the traffic sign detection algorithm starts by choosing the right color model. This detection algorithm faces many challenges due to the outdoor light conditions. It should perform well in different weather conditions and be invariant to shadows and highlights. The choice of the right color model is critical since it determines the level of complexity of the color segmentation algorithm and its performance.

There is a wide range of mathematical models that describes color but only few have a useful meaning for this application. The first color model, that might be a good fit, is the RGB color model. The later describes color in the same way light emitting media and human eyes interact with light. They all have a perception of light as a mixing of additive primary colors Red, Green and Blue. Although the RGB color model has this proximity to the hardware level of light transmitting media and the biological level of the human eye, it is unintuitive to use when processing colorful images. A color in the RGB color model is defined as tuple of red, green and blue, thus the color information can only be extracted using the three components of this tuple combined. This dependency adds a level of complexity when processing colorful images in the RGB color model. The HSV color model solves this dependency problem since the color information is contained in only one component, the hue. This color model's representation of color doesn't fit the functioning of some light transmitters and receptors. However, it matches perfectly how the human brain imagine and think of colors. When imagining a specific color, the first information that is determined is the main color (hue) then the amount of this color

(saturation) and finally the brightness level. Furthermore, the linearly separable color information makes the HSV color model more invariant to shadow and highlight. Consequently, the HSV color model is the perfect choice for the traffic sign detection algorithm.

2.2 The advanced color segmentation

The color segmentation is the second phase of the detection algorithm. After reading the image and projecting it to the HSV color space, the color segmentation extracts regions of specific color as binary masks that define regions of interest. The output of the color segmentation will be processed in the next phase to recognize the shape of the extracted regions.

Color segmentation algorithms usually use a threshold for each component of the color model to filter pixels of a specific color. Using the HSV color model, a threshold on the hue, saturation and brightness will give us an HSV domain of a specific color range (hue), saturation range and brightness range. This approach works in many cases usually in indoor light condition and run faster since only one binary mask, the output of HSV filtering, will be processed. However, this naïve color segmentation fails when dealing with outdoor lighting conditions. Some of the cases where this color segmentation fails is a traffic sign with shadow on it, part of the traffic sign is bright, and the other part is dark, only the bright part will be preserved, and the other part of the sign will be lost. Another case that generates a false negative is when the background of the traffic sign has a slightly similar color as the traffic sign. In this case the building and the traffic will be extracted as

one single object, this extracted region will be deleted in the next shape recognition phase since it doesn't have the right traffic sign shape.

In order to improve the performance of the color segmentation algorithm, an innovative approach was developed in this research project. The advanced color segmentation uses multiple HSV domains with different hue, saturation and brightness (value). Thus, the algorithm uses each HSV domain to extract specific regions and outputs one binary mask per domain for each image. Those HSV domains are very selective, therefore, they enable the color segmentation algorithm to extract a traffic sign even if the background has a slightly similar color. The advanced color segmentation is also shadow invariant. Because the HSV domains are selective, the traffic sign with shadow on it will be separated in two regions, bright region and dark region, and each region will be in a different binary mask. Hence, the algorithm merges the dark binary masks and bright binary masks using an "AND" operator to reconstruct the dark part and bright part of the traffic sign in one single mask. The downside of this technique is the increase of the processing time. The colorful image is processed multiple times for each HSV domain, then the shape recognition algorithm process multiple binary masks for each image.

2.3 Experiment and remarks

The advanced segmentation algorithm is tested on red and blue circular traffic signs as shown in Figure 4 and Figure 5:

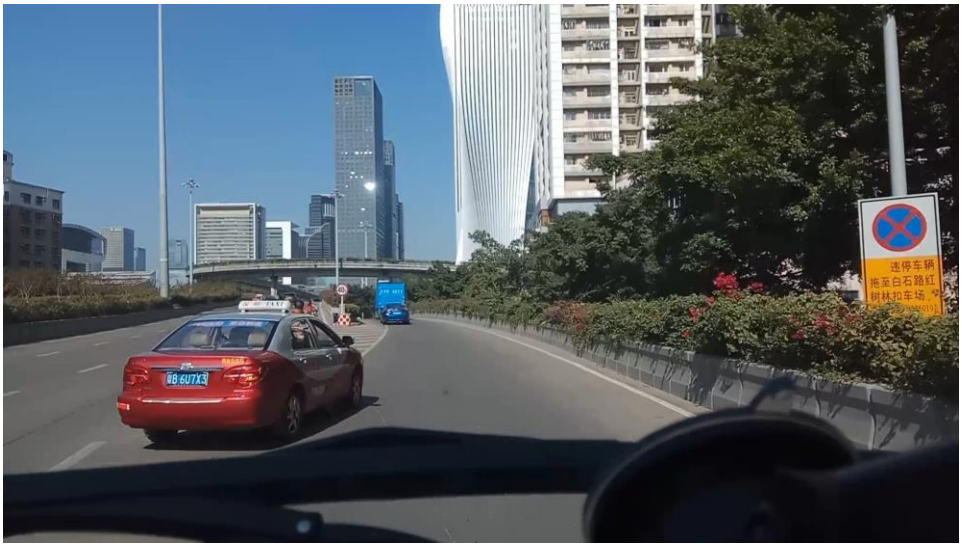

Figure 4: Experimental image with red circular traffic sign

Figure 5: Experimental image with blue traffic sign

2.3.1 Red color

The red circular traffic signs are the most challenging ones because they only have few red pixels. In order to have the lowest false negative score, the color segmentation algorithm for red color uses 17 HSV domains (Table 1). Each HSV domain outputs one binary mask, and one last binary mask is added as the union of all 17 masks using the "AND" operator.

Table 1: HSV domains for red color

	D1	D2	D3	D4	D5	D6	D7	D8
H min	0.915	0.711	0.915	0.865	0.797	0.945	0.028	0.967
H max	0.042	0.061	0.991	0.048	0.005	0.056	0.085	0.086
S min	0.449	0.036	0.11	0.18	0.141	0.154	0.132	0.154
S max	1	0.377	0.393	0.498	0.542	0.71	0.425	0.431
V min	0.417	0.181	0.298	0.596	0.347	0.35	0.599	0.265
V max	1	0.5	0.461	1	0.554	0.681	0.696	0.408

	D9	D10	D11	D12	D13	D14	D15	D16	D17
H min	0.017	0.797	0.853	0.974	0.035	0.951	0.89	0.981	0.709
H max	0.084	0.005	0.039	0.063	0.085	0.038	0.058	0.089	0.934
S min	0.122	0.141	0.259	0.516	0.254	0.195	0.192	0.046	0.247
S max	0.261	0.542	0.619	0.871	0.69	0.498	0.523	0.372	0.605
V min	0.322	0.347	0.151	0.223	0.495	0.596	0.105	0.763	0.141
V max	0.461	0.554	0.407	0.401	0.81	0.902	0.285	1	0.376

The advanced color segmentation algorithm and the naïve color segmentation algorithm were tested on a dataset of 95 traffic signs (Table 2).

Table 2: Performance of advanced and naive color segmentation

	Number of signs	False negative	Detection score
Advanced segmentation	95	3	96.84%
Naïve segmentation	95	12	87.36%

At this stage, using only color segmentation, the two algorithms are compared regarding the number of false negatives because false positives score depends on the shape recognition that is developed in Chapter 3. The advanced segmentation performs better than the naïve one as shown in Table 2.

Figure 6 shows a case where the naïve color segmentation failed to extract the sign because the background had an orange color that was in the range of the single HSV domain

used. Thus, the naïve color segmentation algorithm extracted on region containing the building and the traffic sign.

Figure 6: (top) image of traffic sign with challenging background, (bottom) output mask using naive segmentaton algorithm

The advanced color segmentation uses 17 different and very selective HSV domains. Therefore, it was able to extract the traffic sign from the background using the HSV domain number 11 as shown in Figure 7. The green bounding box around the traffic

sign defines the region of interest detected by the color segmentation. The output mask clearly shows a white isolated circle that correspond to the traffic sign.

Figure 7: (top) image of traffic sign with challenging background detected, (bottom) output mask of advanced segmentation algorithm using HSV domain number 11

The advanced color segmentation was also able to extract the orange building using the HSV domain number 8 and 16 as shown in Figure 8. The advanced color segmentation algorithm has HSV domains able to extract the orange/red color because the red traffic sign might have some orange/red pixels in some lighting conditions. However, the regions of

interest that correspond to the building will be filtered using shape recognition in the next detection phase.

Figure 8: (top) output mask of the advanced segmentation algorithm using HSV domain number 8, (bottom) output mask of the advanced segmentation algorithm using HSV domain number 16

Although the advanced color segmentation algorithm scores a high detection rate, it stills fails in some cases. The image in Figure 9 is a case where the color segmentation was not able the reconstruct the sign in any binary mask as shown in Figure 10.

Figure 9: Image of traffic sign, false negative case using the advanced segmentation algorithm

Figure 10: 18 binary masks, the color segmentation result of image in Figure 9

2.3.2 Blue color

Unlike red traffic signs, the blue traffic signs are easy to detect since they have an important number of pixels. The color segmentation algorithm for blue color uses only 3 HSV domains as shown in Table 3. Similarly to the red color segmentation algorithm, each HSV domain outputs one binary mask, and one last binary mask is added as the union of all 17 masks using the "AND" operator.

Table 3: HSV domains for blue color

	D1	D2	D3
H min	0.578	0.597	0.58
H max	0.617	0.64	0.613
S min	0.638	0.679	0.59
S max	0.985	0.897	1
V min	0.448	0.295	0.408
V max	0.645	0.537	0.593

The blue color segmentation algorithm was tested on 54 signs and was able to detect 100% of them. As example of blue color segmentation, the image in Figure 11 and Figure 12 was processed using three HSV domains defined above and in this case the three HSV domains were able to extract the sign. In the first image (Figure 11), the rectangular blue traffic sign was detected by the all three HSV domains while in the second image (Figure 12) the circular traffic sign at the left was detected by the HSV domain number 2 and the union binary mask.

Figure 11 : (top) first image with blue traffic sign, (bottom) 4 binary masks of the color segmentation of the image at the top

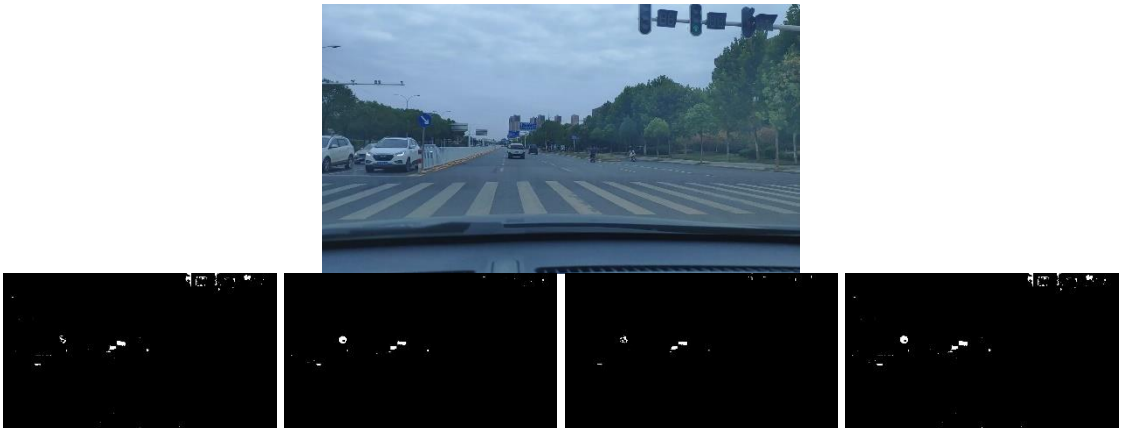

Figure 12: (top) second image with blue traffic sign, (bottom) 4 binary masks of the color segmentation of the image at the top

Figure 13: false positive example, (top) third image with blue traffic sign, (bottom) 4 binary masks of the color segmentation of the image at the top

The blue color segmentation algorithm scores a high detection rate. However, the algorithm extracted many regions that will potentially be false positives after the shape recognition because those regions look the same as a traffic sign, they are blue and rectangular as shown in Figure 13.

CHAPTER 3 SHAPE RECOGNITION

3.1 Introduction

The shape recognition is the last phase of the detection algorithm, it aims to recognize the shape of the extracted regions. The color-based approach is not enough to detect traffic signs because there are objects with the same color as traffic signs like cars, building, etc. However, the color segmentation gives regions of interest, so instead of processing the whole image using a shape-based algorithm, which can be very challenging, only regions of interest of the image are processed. The shape detection algorithm is applied to regions of 1-bit binary images, so low computational resources is required.

The traffic signs have a defined and known shape for each color. This research project covers traffic signs with circular, rectangular and triangular shapes. For circular traffic signs, the shape detection algorithm should detect elliptical shapes because circular shapes become elliptical due to the projected perspective. If an image is not taken in front of the traffic sign which means the image plane is not parallel to the plane of the traffic sign, which is usually the case, the projection of the circular shape on the image plane is an ellipse. Therefore, the shape recognition detects ellipses and not circles.

The images are taken from random distances, angles and orientations. Thus, the shape recognition algorithm should be invariant to scale and rotation, so classic algorithms using template matching cannot be used. The recognition should also be fast in order to achieve a real time processing. Therefore, feature extraction algorithms like Hough transform or feature detection and matching algorithms like SIFT and RANSAC will not run fast enough.

3.2 The algorithm

The shape recognition algorithm uses affine moment invariant to compute ellipticity and triangularity and it uses the standard minimum bounding rectangle method to measure rectangularity. Then, it combines the three shape measures to classify each region of interest into four categories {elliptical, rectangular, triangular, Nan}.

To measure ellipticity and triangularity, the shape recognition process starts by filling the holes of the shapes, then it uses moment invariant to measure the ellipticity and triangularity of the shape. First, it computes the second order central moments μ_{02}, μ_{02}, the first order central moment $\mu_{11}$and the zero-order central momentμ_{00}.

The affine central moments are invariants to affine transformation applied to the image. An affine transformation is a linear transformation to the coordinate system of the image:

$$u = a_u + b_u x + c_u y$$

$$v = a_v + b_v x + c_v y$$

Affine transformations, like scaling and rotating, doesn't change the value of affine central moments, which makes this measure invariant to scale and rotation. Raw moment of order $(p + q)$ applied to a 2D image, $(x, y) \rightarrow I(x, y)$, is defined as:

$$M_{pq} = \sum_x \sum_y x^p y^q I(x, y)$$

Using the raw moment definition, two properties can be derived:

- M_{00} is the area of the region for binary images
- The centroid or geometric center is $\{\bar{x}, \bar{y}\} = \left\{\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right\}$

Central moments are computed in terms of deviations from the mean, unlike the raw moment that are computed in terms of deviation of from zero. In image processing applications, this means that central moments are normalized with respect to translation. A central moment of order $(p + q)$ applied to a 2D image, $(x, y) \rightarrow I(x, y)$, is defined as:

$$\mu_{pq} = \sum_{x}\sum_{y}(x - \bar{x})^p (y - \bar{y})^q I(x, y)$$

$where\ \bar{x}\ and\ \bar{y}\ are\ to\ component\ of\ the\ centroid$:

$$\bar{x} = \frac{M_{10}}{M_{00}}, \bar{y} = \frac{M_{01}}{M_{00}}$$

The ellipticity and triangularity measure use the affine moment invariant I_1 [10]:

$$I_1 = \frac{(\mu_{20}\mu_{02} - \mu_{11}^2)}{\mu_{00}}$$

To discriminate shapes more precisely higher order invariants should be involved. However, they are less reliable and very sensitive to the noise. In contrast I_1 is stable and more practical to use.

To measure the ellipticity E, the following equation is used [10]:

$$E = \begin{cases} 16\pi^2 I_1 & if\ I_1 \leq \dfrac{1}{16\pi^2} \\ \dfrac{1}{16\pi^2 I_1} & otherwise \end{cases}$$

For a perfect ellipse E=1, since the ellipse of the traffic sign is not perfect, a threshold (<1) should be used.

To measure the triangularity T, the following equation is used [10]:

$$T = \begin{cases} 108\ I_1 & if\ I_1 \leq \dfrac{1}{108} \\ \dfrac{1}{108\ I_1} & otherwise \end{cases}$$

For a perfect triangle T=1, but a threshold (<1) should be used to consider imperfect triangles with round corners or curved edges.

Rectangularity is measured using standard minimum bounding rectangle approach. The minimum bounding rectangle (MBR) of an object is the smallest bounding box that the object fits in, as shown in Figure 14.

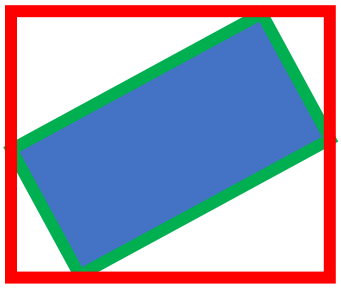

Figure 14:Minimum bounding box (green), normal bounding box (red), region of interest (blue)

The rectangularity measure R is defined as the ratio of the area of the object to the area of its minimum bounding rectangle (MBR) [10]:

$$R = \frac{area(Object)}{area(MBR)}$$

For a perfect rectangle R=1, but like triangularity measure a threshold (<1) should be used to consider imperfect rectangles with round corners or curved edges.

There is a trade of, between the number of false positives or accuracy and the number of false negatives or recognition rate, that is made when choosing the right threshold. By using a low threshold, the shape recognition algorithm will generate less false negatives and more false positives, which means that the shape recognition is less accurate but has high recognition rate. On the other hand, using a low threshold, generates more false negatives and less false positives, which means that the accuracy is better, but the algorithm recognizes less shapes than before. After running the shape recognition algorithm on multiple datasets, the optimal threshold used is 0.98, refer to APPENDIX A for more detail about the dataset used.

The shape recognition algorithm preprocesses the binary mask that contains the region of interest before measuring the shape. For the ellipticity and triangularity measure, the algorithm fills the holes of the region of interest, making the region solid, before computing the central moments, the affine moment invariant then the ellipticity and triangularity score, as shown in Figure 15.

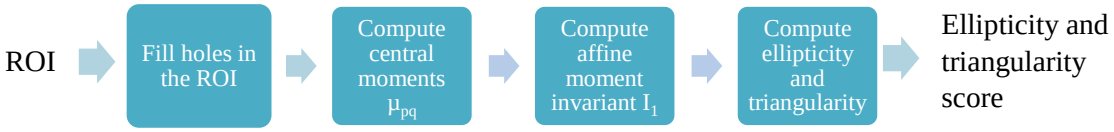

Figure 15:Process to compute the ellipticity and triangularity measures

To compute the rectangularity measure, the algorithm starts by computing the MBR of the region of interest, then it computes the ratio of the area of the ROI to the area of its MBR to outputs the rectangularity measure, as shown in Figure 16.

Figure 16: Process to compute the rectangularity measure

3.3 Experiment

The performance of the shape recognition algorithm depends on the accuracy and the discriminative aspect of the shape measures. As shown in Figure 17, the shape measures for ellipses, rectangles and triangles remain accurate and discriminative at different aspect ratio.

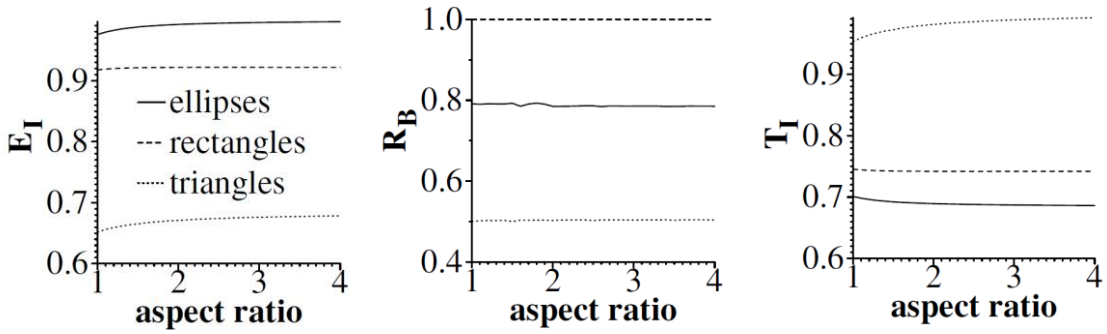

Figure 17: Ellipticity E_I, triangularity T_I and rectangularity R_B at different aspect ratio (from [10])

In order to evaluate the discriminative force of each measurement, the ellipticity, triangularity and rectangularity are drawn as a function of the squareness parameter, as

shown in Figure 19. The relation between the squarenes parameter and shapes is shown at Figure 18. It means that the ellipticity, triangularity and rectangularity are measured for different shapes of Figure 18, a first time ranging from square to circle, and a second time ranging from triangle to circle. Analysing the data of Figure 17 and Figure 19 shows that an algorithm that combines the three measures to classify the shapes would be performant and accurate:

- An ellipse has a very high value of ellipticity, a high value of rectangularity and a very low value of triangularity
- A rectangle has a very high value of rectangularity, a high value of ellipticity and a very low value of triangularity
- A triangle has a very high value of triangularity, a very low value of ellipticity and a very low value of rectangularity

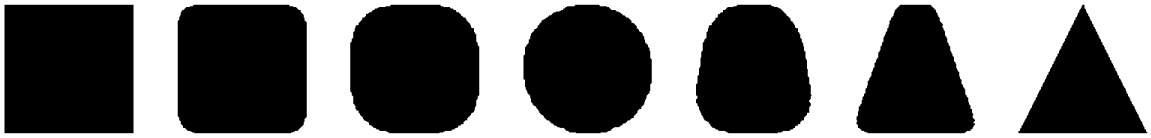

Figure 18: Shapes from square to circle and from triangle to circle (from [10])

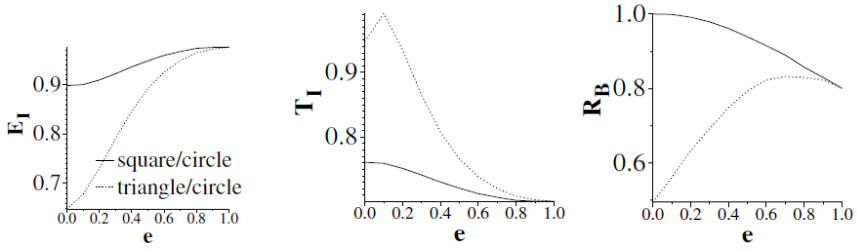

Figure 19: Ellipticity E_I, triangularity T_I and rectangularity R_B with regards to the squareness parameter e (=0 for square/triangle =1 for circle) (from [10])

The performance of shape recognition algorithm depends also on the preprocessing of the region of interest. If the preprocessing fails, the measure will be corrupted. For the ellipticity and triangularity measures, the region of interest is filled, but in order to fill the region of interest, it should be a closed region. An open region like a broken ring break the filling algorithm. When dealing with red circular traffic sign, the shape recognition algorithm fails if the red circle is broken and open. Similarly, the triangle should be closed in order to fill the holes inside the triangle.

The shape recognition algorithm was tested on 96 circular traffic signs, 53 rectangular traffic signs and 17 triangular traffic signs. A false negative in this evaluation means that the shape recognition was unable to classify a region of interest or misclassified it.

Table 4: Performance of the shape recognition algorithm

	Number of signs	Number of false negative	Recognition score
Circular	93	2	97.84%
Rectangular	53	1	98.11%
Triangular	17	5	70.58%

The performance of the shape recognition algorithm depends on the output of the color segmentation phase. Although the recognition score is high, it still fails when the region of interest is not a closed region, especially for ellipses and triangles. The recognition score of the triangularity measure is low because the dataset contains many cases of triangle that is not a closed region, so the hole filling algorithm failed.

The traffic sign in Figure 20 is a speed limit sign, with a red circle. The advanced color segmentation algorithm was able to extract the red pixels of the traffic sign using the HSV domain number 1, as shown in the binary mask of Figure 20. However, the extracted circle is open at two different positions. The hole filling algorithm failed while processing this region of interest, which corrupted the ellipticity, triangularity and rectangularity measures.

Figure 20: Example of a broken circle, (top) the image of a red speed limit traffic sign, (bottom) the output mask of the advanced color segmentation using the HSV domain number 2

CHAPTER 4 PARALLEL COMPUTING

4.1 Multi-processing and multi-threading

When dealing with a big amount of data, multi-processing can be the key for a fast processing of this data. In many cases, if the data is not processed before a deadline, the resulted output is useless. This is the case for embedded systems for example, an autonomous driving car processes data coming from cameras LIDARs and other sensors, and it must process it fast enough in order to use the output to correctly drive the vehicle by changing the speed and the orientation of the car.

Multi-processing is a type of parallel computing. This type of computation executes multiple processes in parallel and simultaneously [11]. The multi-processing computation depends on the hardware and its performance is limited by the number of the processing units available.

A single core CPU (central processing unit) can only execute one task at time, which means that two processes cannot run simultaneously. Although a single core CPU might give the impression that processors are running in parallel, it is just using concurrent computing which should not be mistaken for parallel computing [12]. Concurrent computing runs processes concurrently during overlapping time periods, which means that computations are executed out of order or in partial order without affecting the result, unlike a sequential execution where one process completes before the next one starts. Time-sharing multitask computation on a single core CPU is an example of concurrent computing. Although concurrent computing enables multiple processes to run in pseudo "parallel", it is less powerful than a real parallel computation. Multithreading is the ability

of a processing unit to execute processes concurrently. The advantage of using multithreading is that multithreading programs have a low memory footprint and has a shared memory between the threads of the same process [13], which enable threads to easily communicate and send object using the address of the object in the shared memory. However, multithreading computation has the same computation limits as concurrent computing, only one core of the CPU or one single core CPU is used, which significantly limits the computation speed. The shared data in multithreading computation has some disadvantages, the thread scheduling algorithm swap between threads randomly, thus threads might want to access the same part of the shared memory without following any order. Therefore, threads are racing to access the data and change it, which may corrupt the data, that's why multithreading computation increases the probability of race conditions.

Multiprocessing uses multiple computation units, in a multicore and multiprocessors computer, to execute multiple computations in parallel and at the same time, which increase the processing speed significantly. In multiprocessing computation, each process has it own virtual memory which eliminates the race conditions and reduce the needs for synchronization primitives regarding memory access. However, multithreading computation has a larger memory foot print than multithreading. Also, a good parallel programing performance depends on a fluid communication and synchronization between different processes, and that's the biggest challenge when coding a program that uses parallel computing.

In multiprocessing computation, processors need a protocol to communicate between each other, they can't send and receive objects using the address of that object in

memory because each process has its own virtual memory and cannot easily access the virtual memory of another process. Therefore, the message passing interface (MPI) is widely used in parallel computing applications and high-performance computing. It is a message passing standard that supports C, C++ and Fortran and enables processes to communicate between each other, by either using a point-to-point or collective communication.

4.2 Architecture of the multi-processing algorithm

The traffic signs detection algorithm needs to run real time, but the advanced color segmentation and the shape recognition involves some heavy computations. In order the decrease the processing time, a multiprocessing algorithm is used to take benefit of the full power and capabilities of the multicore CPU.

The algorithm uses MPI to establish a communication between several processes in use. The architecture of the algorithm consists of one master process that communicates with all the other processes and give them the right frame number to process. The master process job is to keep track of which processors is processing which frame, it should also give the processors the name of the next frame to process and receives the result of the last frame they processed. The other processes' job is to process the frame and communicate the output with the master process.

Assuming the frames are named from 0 to (numberOfFrames-1), during the first iteration, the master process, whose rank is 0, sends the other processes of rank i>1 the number (i-1) so that each process will execute the traffic sign detection algorithm on frame

number (i-1). After the first iteration, the master process will execute the steps below, in a loop, until all the frames are processed.

- Step1: Wait for a message from any process
    - Receive messages from the same process until you receive -1
    - Save the result received
- Step2: Send the next frame number to the process that just sent the result or send -1 if all frames are processed

The other processes send 4 integers at time to the master process. The 4 integers are the x, y, height and width of the region of interest detected. If the process detects N region of interests, it will send N times 4 integers and it will send 4 integers of -1 at the end to notify the master process that region of interest was sent. The Master process will keep listening to that same process until receiving 4 integers all equal to -1. If the process detects only one region of interest, it will send two messages, the first one is the region of interest (x,y,height,width) and the second one is (-1,-1,-1,-1).

The other processes will execute the steps below starting from the first iteration, in a loop, until they receive -1 from the master process, which means that all the frames were processed, then they will end themselves.

- Step1: Wait for a message from the master process
    - If the number is -1, ends the process

- Else, save the frame number received

- Step 2: Process the frame received

- Step3: Send the result to the master process

The multiprocessing algorithm was tested using different number of cores and was able to significantly speed up the detection algorithm as shown in Table 5.

Table 5: Processing time of the traffic sign detection algorithm using multiprocessing

Number of process	Processing time (s)	Number of frames processed
3	75.761	376
4	50.318	376
5	37.660	376
6	30.096	376
7	25.054	376
8	21.456	376

In the current architecture, one process doesn't execute the traffic sign detection algorithm and handles the communication and the storage of the result, if N process are used, only N-1 process frames.

Theoretically, the traffic sign detection algorithm using multiprocessing should run N-1 times faster than the standard algorithm ran on a single core. As shown in Figure 21, the multiprocessing algorithm runs nearly N-1 times faster than the single process algorithm.

Unfortunately, the communication adds a delay to the processing time, but the detection algorithm still runs faster and real time using the multiprocessing computation.

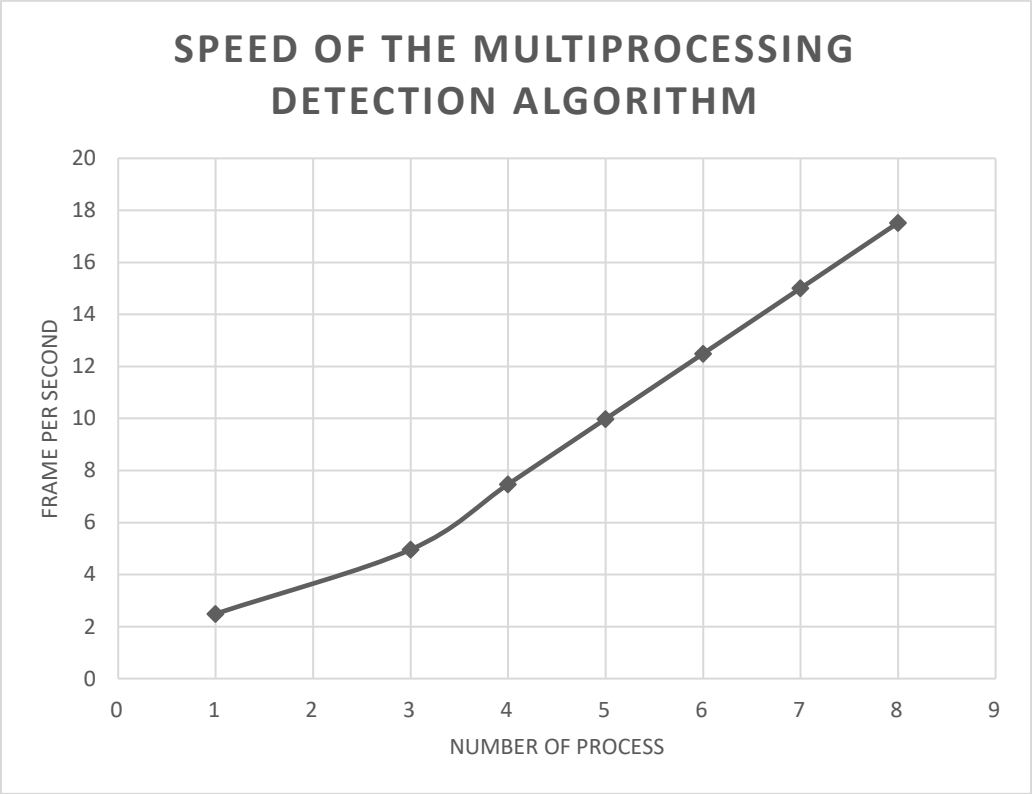

Figure 21: speed of the traffic sign multiprocessing detection algorithm, number of frames processed per second as function of number of process used

CHAPTER 5 THE COMPLETE ALGORITHM

The final traffic sign detection algorithm is described in Figure 22.

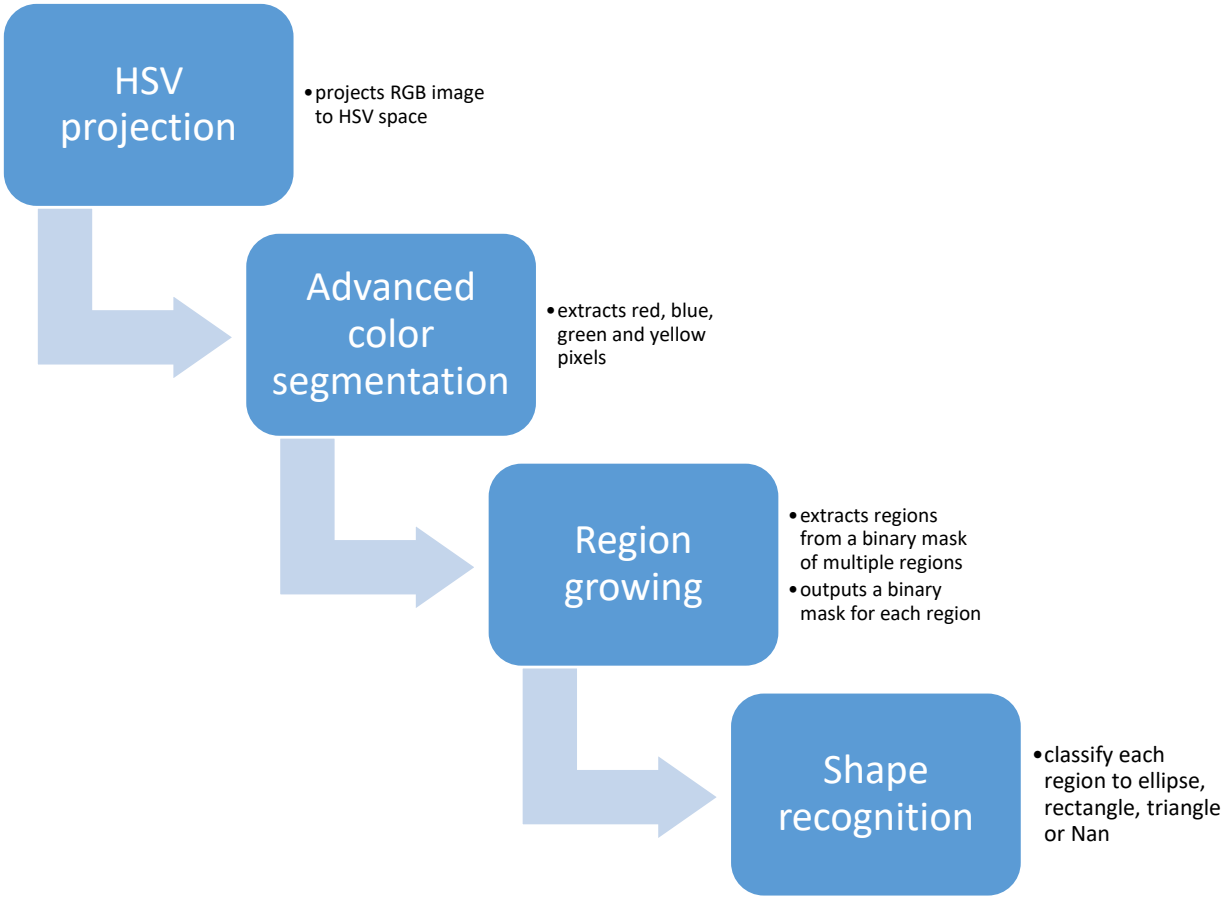

Figure 22: The pipeline of the traffic sign detection algorithm

The region growing process is the third phase of the traffic sign detection algorithm. It takes the binary masks of the advanced color segmentation as input. Each mask from the color segmentation process contains several regions but the shape recognition algorithm only processes masks that contains one region. Thus, the region growing algorithm extracts the regions and outputs one binary mask for each region.

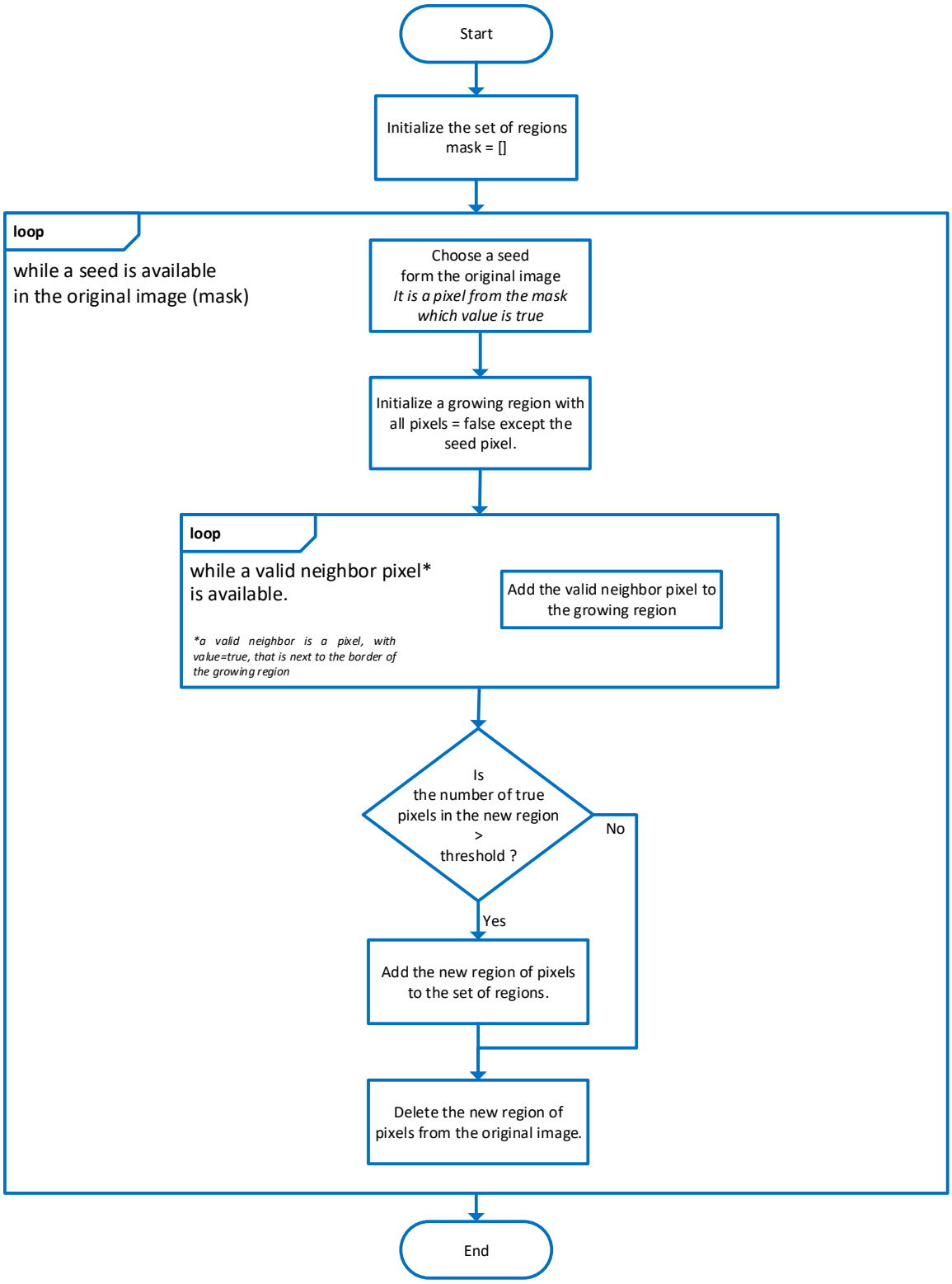

Figure 23: Interaction overview UML diagram of the traffic sign detection algorithm

The traffic sign detection algorithm is tested on 149 signs as shown in Table 6.

Table 6: Performance of advanced and naive color segmentation

	Number of signs	False Positive	False negative	Accuracy	Detection score
Detection using advanced segmentation	149	2	4	98.62%	97.31%
Detection using naïve segmentation	149	14	25	88.70%	83.22%

This test shows that the advanced color segmentation algorithm has a better performance, it is 9.92% more accurate and detects 14.09% more signs than the naïve color segmentation algorithm.

CHAPTER 6 CONCLUSION AND FUTURE WORK

The traffic sign recognition algorithm was able to achieve a good performance on very challenging datasets. The advanced color segmentation algorithm increased significantly the detection rate and the accuracy. It was able to extract traffic signs that suffers from shadow and highlight. The use of HSV space made the algorithm immune to the change of many weather and light conditions. The shape recognition algorithm was able to eliminate a huge number of false positives and made the detection possible. Its invariance to scale and rotation make it robust and reliable. The multiprocessing algorithm made it possible to achieve a high processing speed. Using 8 cores the traffic sign detection algorithm can process 17.53 frames per second.

The traffic sign detection algorithm still faces some challenges at the color segmentation level and shape recognition level. The current color segmentation algorithm is still unable to reconstruct some colors, that are lost due to the limited color range of the camera, and extract regions with missing parts and that are not closed. This issue generates a problem for the shape recognition algorithm which cannot fill the holes of regions if they are not completely closed. Although applying dilation then erosion can close some regions, but this approach fails when the opening is large.

Finally, using a dedicated hardware for parallel computing like GPU should improve the speed of the algorithm. GPUs are very efficient at processing images and have a hardware architecture designed for parallel computing.

APPENDIX A. RESULT OF SHAPE RECOGNITION

The table below shows the ellipticity score for 327 regions of interest used to define an optimal threshold to be used in the traffic sign detection algorithm. It shows the HSV domain that was used to extract that region of interest.

Table 7: Ellipticity score for 200 regions of interest

FRAME NUMBER	ELLIPTICITY SCORE	HSV DOMAIN	ROI SIZE	ROI FILLED SIZE
59035	0.995078	17	735	1786
59035	0.998063	17	687	1721
59035	0.996516	17	857	1733
94900	0.995401	3	1697	4743
94900	0.995543	15	2486	5129
94900	0.996629	17	2782	6070
94900	0.990496	17	2964	5459
94910	0.999419	0	4419	14866
94910	0.986674	3	3518	11356
94910	0.999254	5	4419	15262
94910	0.99648	15	2525	12486
94910	0.99782	17	6603	15902
94910	0.995372	17	5908	12814
12698	0.977305	5	9747	24292
12698	0.997671	5	10043	22226
12698	0.96334	10	7060	21866
12698	0.995439	11	6237	20073
74404	0.990069	17	2880	4588
40051	0.999822	0	2888	10003
40051	0.996048	5	3367	10483
40051	0.996799	17	5651	11577
62340	0.991781	3	950	3181
62340	0.994778	3	827	2740
62340	0.991188	17	1455	3322
62340	0.989731	17	1234	2897
94526	0.999723	3	3621	9749
94526	0.999371	13	3259	9576
94526	0.999434	15	1203	6463
94526	0.998748	17	4601	9861

95167	0.994609	17	2595	6540
14303	0.998806	1	3049	7034
14303	0.998872	1	3188	6970
14303	0.999144	1	3120	6956
14303	0.998952	1	3333	9482
14303	0.993096	4	1603	6346
14303	0.993096	9	1603	6346
14303	0.998046	10	1500	6318
14303	0.998376	10	2031	6498
14303	0.999455	10	2204	6555
14303	0.996066	10	3346	8710
14303	0.994499	14	2996	9040
14303	0.956932	16	2597	6842
14303	0.998806	17	3049	7034
14303	0.998872	17	3188	6970
14303	0.999141	17	3375	6957
14303	0.997926	17	5365	9610
76437	0.999196	10	5273	16718
76437	0.998113	17	7384	17695
44772	0.996591	1	1448	3187
44772	0.997559	17	1513	3223
52701	0.999868	0	3513	12478
52701	0.997682	17	6460	14018
62125	0.999079	4	465	2376
62125	0.999079	9	465	2376
62125	0.99528	17	1350	2641
88448	0.993087	16	1673	3492
88448	0.991928	17	1785	3604
62339	0.995737	3	921	3068
62339	0.990619	17	1314	3216
59273	0.96982	17	3899	5219
62352	0.999295	3	2732	8280
62352	0.999059	3	2347	6800
62352	0.976391	5	1723	7449
62352	0.998611	13	2470	8082
62352	0.998122	13	2030	6585
62352	0.995922	17	3727	8604
62352	0.996527	17	3067	6946
33199	0.98411	15	3765	9466
33199	0.984902	15	5123	8715
33199	0.984769	15	3688	7970

33199	0.988424	15	4271	8070
45921	0.999728	1	1549	2239
45921	0.988296	4	755	2172
45921	0.988296	9	755	2172
45921	0.999908	10	664	2139
45921	0.999124	17	1610	2284
14302	0.999084	1	2713	6480
14302	0.999103	1	2999	6360
14302	0.999154	1	3124	6328
14302	0.999254	1	2851	8612
14302	0.997557	2	1934	6116
14302	0.98659	2	1634	5770
14302	0.968897	4	1689	5500
14302	0.968897	9	1689	5500
14302	0.998915	10	1850	5919
14302	0.999394	10	1987	5958
14302	0.995749	10	2563	7958
14302	0.991358	14	2694	8304
14302	0.971541	16	2373	6346
14302	0.999084	17	2715	6480
14302	0.999103	17	2999	6360
14302	0.999147	17	3129	6333
14302	0.997641	17	4784	8757
45569	0.999294	4	2267	5759
45569	0.999294	9	2267	5759
45569	0.999734	10	1963	5577
45569	0.998677	17	3372	5919
62121	0.999307	4	445	1908
62121	0.999307	9	445	1908
62121	0.994569	17	1225	2091
40053	0.999655	0	3849	12768
40053	0.992205	5	4552	13380
40053	0.974072	17	7649	15054
74397	0.98218	17	2041	3485
94905	0.99923	0	2284	8479
94905	0.998677	3	2495	7068
94905	0.997803	5	2942	8848
94905	0.997701	15	3415	7613
94905	0.996147	17	4144	9516
94905	0.995158	17	3757	7871
94524	0.999521	0	3496	11024

94524	0.999671	3	3081	8141
94524	0.999818	3	671	7604
94524	0.999541	13	2763	7987
94524	0.996178	15	983	5408
94524	0.998993	17	3814	8231
27136	0.999154	1	3728	8927
27136	0.999494	1	3213	8446
27136	0.999212	1	2921	8165
27136	0.997518	2	2728	8487
27136	0.997282	4	2697	8431
27136	0.997029	4	2946	8029
27136	0.996816	5	2780	7984
27136	0.986196	5	2324	7212
27136	0.997282	9	2697	8431
27136	0.997029	9	2946	8029
27136	0.998844	10	2032	8181
27136	0.999373	10	2414	7958
27136	0.999434	10	2583	7777
27136	0.999149	17	3731	8930
27136	0.999531	17	3599	8451
27136	0.999227	17	3744	8171
12699	0.996156	5	11074	24939
12699	0.98833	11	7642	22999
95168	0.993579	3	1885	6511
95168	0.996871	15	1866	4048
95168	0.996831	17	2577	6821
59044	0.999215	1	602	3278
59044	0.9994	4	698	3296
59044	0.997896	5	846	3465
59044	0.998609	5	827	3336
59044	0.997907	5	755	3207
59044	0.997897	5	785	3160
59044	0.9994	9	698	3296
59044	0.996516	15	507	2253
59044	0.999256	17	1480	3663
59044	0.999288	17	1588	3476
59044	0.9995	17	1588	3363
59044	0.999336	17	1553	3299
64123	0.999676	0	1322	6195
64123	0.999494	3	2421	6832
64123	0.99731	13	1945	6486

64123	0.992277	15	915	4719
64123	0.999493	17	3038	6836
32697	0.9974	1	1318	5429
32697	0.999724	4	1546	5507
32697	0.999724	9	1546	5507
32697	0.99979	17	1640	5542
88623	0.954017	10	1153	2833
94907	0.998948	0	3016	10517
94907	0.998277	3	2956	8447
94907	0.997834	5	3847	11044
94907	0.996656	15	2944	9153
94907	0.996718	17	5204	11682
94907	0.993236	17	4411	9429
88625	0.980246	10	1194	3055
33198	0.990877	15	3845	9181
33198	0.99345	15	4671	8306
33198	0.985887	15	4191	7766
33198	0.98881	15	4164	7649
33198	0.985237	15	2570	7263
63662	0.997369	3	1051	2972
63662	0.990845	17	1347	3268
62334	0.984958	17	1052	2389
62334	0.990526	17	958	2150
74202	0.999117	4	615	2131
74202	0.999117	9	615	2131
74202	0.997433	17	698	2211
32699	0.997511	1	1624	6292
32699	0.999576	4	1808	6361
32699	0.999576	9	1808	6361
32699	0.999732	17	1946	6434
62326	0.979716	17	625	1605
62326	0.981508	17	710	1572
45572	0.996999	5	3023	7667
45572	0.989273	10	2206	7297
45572	0.999432	17	4871	7826
62351	0.999046	3	2578	7604
62351	0.998972	3	2107	6185
62351	0.998052	13	2219	7301
62351	0.995913	13	1740	5903
62351	0.995499	17	3443	7877
62351	0.996323	17	2756	6362

59039	0.999007	1	459	2437
59039	0.997743	1	451	2339
59039	0.998963	1	447	2280
59039	0.99905	1	503	2255
59039	0.999044	17	896	2438
59039	0.997843	17	981	2340
59039	0.998988	17	993	2281
59039	0.99927	17	951	2259
93958	0.99974	0	2338	9200
93958	0.994037	3	3605	10130
93958	0.978757	5	3560	10302
93958	0.994615	13	3129	9819
64122	0.999455	3	2184	6266
64122	0.997883	13	1664	5916
64122	0.98196	15	1210	4557
64122	0.999274	17	2963	6275
94902	0.997745	0	1611	6347
94902	0.99761	3	1905	5454
94902	0.988445	5	2307	6789
94902	0.997463	15	2730	5942
94902	0.990038	17	3342	7321
94902	0.993225	17	2994	6192
74403	0.985289	17	2837	4484
62344	0.997493	3	1405	4339
62344	0.998809	3	1254	3722
62344	0.991599	5	1173	4116
62344	0.994652	13	1144	4101
62344	0.995779	13	964	3479
62344	0.993198	17	2053	4534
62344	0.995263	17	1644	3817
6957	0.998727	0	3598	10924
59043	0.999241	4	604	3022
59043	0.99837	5	736	3060
59043	0.997225	5	672	2860
59043	0.999241	9	604	3022
59043	0.999421	17	1375	3396
59043	0.999043	17	1419	3183
59043	0.99938	17	1390	3100
59043	0.999221	17	1284	3000
74414	0.978065	17	5336	8884
59040	0.998784	1	517	2635

59040	0.998692	1	535	2393
59040	0.997243	4	495	2540
59040	0.997243	9	495	2540
59040	0.998784	17	973	2635
59040	0.99858	17	987	2455
59040	0.998764	17	1043	2423
59040	0.998962	17	1109	2404
59046	0.99917	1	542	3822
59046	0.993258	3	679	3360
59046	0.999305	4	776	3854
59046	0.997753	5	1063	4104
59046	0.998518	5	1015	3911
59046	0.998975	5	991	3813
59046	0.99951	5	1047	3776
59046	0.999305	9	776	3854
59046	0.999806	15	680	2674
59046	0.999332	17	1838	4316
59046	0.999355	17	1710	4053
59046	0.999476	17	1510	3924
59046	0.999284	17	1852	3846
74335	0.997164	3	1666	4130
74335	0.99081	5	1707	4548
74335	0.997305	13	1532	4089
74335	0.982771	17	3058	5060
63668	0.997953	3	1516	4044
63668	0.995304	17	1963	4491
14304	0.997424	1	3402	7685
14304	0.982572	1	3266	7405
14304	0.998828	1	4044	10303
14304	0.997252	2	2484	7405
14304	0.994751	4	2009	7142
14304	0.994751	9	2009	7142
14304	0.998374	10	1947	7123
14304	0.995776	10	2269	7184
14304	0.995626	10	3811	9599
14304	0.992894	14	3415	9874
14304	0.997424	17	3402	7685
14304	0.993954	17	3520	7593
14304	0.997949	17	5767	10457
74334	0.996333	3	1570	3863
74334	0.991903	5	1588	4223

74334	0.995884	13	1429	3823
74334	0.992061	17	2811	4695
74416	0.992253	17	5795	9988
64114	0.999104	3	1091	3357
64114	0.999104	17	1548	3357
63660	0.996234	3	1030	2753
63660	0.977126	17	1341	3064
63663	0.997028	3	1225	3207
63663	0.991479	17	1539	3521
30826	0.99568	1	3861	4487
30826	0.989126	5	1821	4093
30826	0.978372	10	1752	4377
30826	0.960894	17	4859	5485
62341	0.997746	3	1145	3524
62341	0.99198	5	957	3364
62341	0.99537	13	982	3395
62341	0.993952	17	1653	3644
62337	0.996278	3	808	2680
62337	0.99447	3	745	2411
62337	0.992631	17	1237	2814
62337	0.992376	17	1261	2619
62123	0.998421	3	792	2342
62123	0.998685	4	422	2167
62123	0.998685	9	422	2167
62123	0.995519	17	1281	2392
52702	0.99991	0	4537	15927
52702	0.999763	3	709	11486
52702	0.999658	15	1340	17472
52702	0.999819	15	1398	11197
52702	0.998285	17	7880	17679
62331	0.984994	17	805	2021
62331	0.982687	17	906	1976
15640	0.997691	1	3274	8618
15640	0.998005	1	3096	8246
15640	0.976879	1	2572	7907
15640	0.998028	10	2486	7708
15640	0.990028	14	2046	7409
15640	0.998081	16	1993	8106
15640	0.998524	16	2368	7898
15640	0.997692	17	3283	8620
15640	0.998057	17	3182	8250

15640	0.998036	17	3496	8182
64124	0.999615	0	1427	6636
64124	0.999469	3	2594	7318
64124	0.99904	13	2177	7048
64124	0.988085	15	969	5121
64124	0.999469	17	3166	7318
62120	0.993793	17	1183	2016
94909	0.999508	0	3856	13033
94909	0.999263	5	4205	13349
94909	0.998012	15	2623	11145

REFERENCES

[1] Aïnouz, S., Zallat, J., De Martino, A., & Collet, C. (2006). Physical interpretation of polarization-encoded images by color preview. *Optics Express, 14*(13), 5916-27.

[2] Cheng-Yen Chiang, Kun-Shan Chen, Chih-Yuan Chu, Yang-Lang Chang, & Kuo-Chin Fan. (2018). Color Enhancement for Four-Component Decomposed Polarimetric SAR Image Based on a CIE-Lab Encoding. Remote Sensing, 10(4), .

[3] Huang, C. L., & Hsu, S. H. (2000). Road sign interpretation using matching pursuit method. Image Analysis and Interpretation, 2000. Proceedings. 4th IEEE Southwest Symposium, 2000, 202-206.

[4] Sandoval, H., Hattori, T., Kitagawa, S., & Chigusa, Y. (2000). Angle-dependent edge detection for traffic signs recognition. Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE, 308-313.

[5] Ghica, Si Wei Lu, & Xiaobu Yuan. (1995). Recognition of traffic signs by artificial neural network. Neural Networks, 1995. Proceedings., IEEE International Conference on, 3, 1444-1449.

[6] Shadeed, W., Abu-Al-Nadi, D., & Mismar, M. (2003). Road traffic sign detection in color images. Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on, 2, 890-893.

[7] Fang, C., Yen, S., Fuh, P., & Chen. (2003). A road sign recognition system based on dynamic visual model. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, I/750

[8] Ohara, H., Nishikawa, I., Miki, S., & Yabuki, N. (2002). Detection and recognition of road signs using simple layered neural networks. Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference on, 2, 626-630.

[9] Miura, J., Kanda, T., & Shirai, Y. (2000). An active vision system for real-time traffic sign recognition. Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE, 52-57

[10] Rosin, P. (2003). Measuring shape: Ellipticity, rectangularity, and triangularity. Machine Vision and Applications, 14(3), 172-184.

[11] Omiecinski, E. (1990). Highly parallel computing: G S Almasi and A Gottlieb Benjamin/Cummings, Redwood City, CA, USA (1989) 519 pp hardback. Information and Software Technology, 32(5), 383.

[12] "Concurrency is not Parallelism", Waza conference Jan 11, 2012, Rob Pike.

[13] Yang, J., Cui, H., Wu, J., Tang, Y., & Hu, G. (2014). Making parallel programs reliable with stable multithreading. Communications of the ACM, 57(3), 58-69.